Model-Based Reinforcement Learning with Isolated Imaginations

Minting Pan, Xiangming Zhu, Yitao Zheng, Yunbo Wang, Xiaokang Yang, *Fellow, IEEE*

Abstract—World models learn the consequences of actions in vision-based interactive systems. However, in practical scenarios like autonomous driving, noncontrollable dynamics that are independent or sparsely dependent on action signals often exist, making it challenging to learn effective world models. To address this issue, we propose Iso-Dream++, a model-based reinforcement learning approach that has two main contributions. First, we optimize the inverse dynamics to encourage the world model to isolate controllable state transitions from the mixed spatiotemporal variations of the environment. Second, we perform policy optimization based on the decoupled latent imaginations, where we roll out noncontrollable states into the future and adaptively associate them with the current controllable state. This enables long-horizon visuomotor control tasks to benefit from isolating mixed dynamics sources in the wild, such as self-driving cars that can anticipate the movement of other vehicles, thereby avoiding potential risks. On top of our previous work [1], we further consider the sparse dependencies between controllable and noncontrollable states, address the training collapse problem of state decoupling, and validate our approach in transfer learning setups. Our empirical study demonstrates that Iso-Dream++ outperforms existing reinforcement learning models significantly on CARLA and DeepMind Control.

1 Introduction

Humans can infer and predict real-world dynamics by simply observing and interacting with the environment. Inspired by this, many cutting-edge AI agents use self-supervised learning [2, 3, 4] or reinforcement learning [5, 6, 7] techniques to acquire knowledge from their surroundings. Among them, world models [3] have received widespread attention in the field of robotic visuomotor control, and led the recent progress in model-based reinforcement learning (MBRL) with visual inputs [6, 7, 8, 9]. One representative approach called Dreamer [6] learns a differentiable simulator of the environment (*i.e.*, the world model) using observations and actions of an actor-critic agent, then updates the agent by optimizing its behaviors based on future latent states and rewards (*i.e.*, latent imagination) generated by the world model. However, since the observation trajectories are high-dimensional, highly non-stationary, and often driven by multiple sources of physical dynamics, how to learn effective world models in complex visual scenes remains an open problem.

In this paper, we propose to understand the world by decomposing it into *controllable* and *noncontrollable* state transitions, *i.e.*, $s_{t+1} \sim p(\cdot \mid s_t, a_t)$ and $z_{t+1} \sim p(\cdot \mid z_t)$, according to the responses to action signals. This idea is largely inspired by practical scenarios such as autonomous driving, in which we can naturally divide spatiotemporal dynamics in the system into controllable parts that perfectly respond to the actions (*e.g.*, accelerating and steering) and parts beyond the control of the agent (*e.g.*, movement of other vehicles). Decoupling latent state transitions in this way can improve MBRL in three aspects:

- It allows decisions to be made based on predictions of future noncontrollable dynamics that are independent (or indirectly dependent) of the action, thereby improving the performance on long-horizon control tasks. For example, in the CARLA self-driving environment, potential risks can be better avoided by anticipating the movement of other vehicles.
- Modular world models improve the robustness of the RL agent in noisy environments, as demonstrated in our modified DeepMind Control Suite with the time-varying background.
- The isolation of controllable state transitions further facilitates transfer learning across different but related domains. We can adapt parts of the world model to novel domains based on our prior knowledge of the domain gap.

Specifically, we present Iso-Dream++, a novel MBRL framework that learns to decouple and leverage the controllable and noncontrollable state transitions. Accordingly, it improves the original Dreamer [6] from two perspectives: (i) *a new form of world model representation* and (ii) *a new actor-critic algorithm to derive the behavior from the world model.*

1.1 How to learn a decoupled world model?

From the perspective of representation learning, we improve the world model to separate mixed visual dynamics into an action-conditioned branch and an action-free branch of latent state transitions (see Fig. 1). These components are jointly trained to maximize the variational lower bounds. Besides, the action-conditioned branch is particularly optimized with *inverse dynamics* as an additional objective function, that is, to reason about the actions that have driven the "controllable" state transitions between adjacent time steps.

Nonetheless, as we have observed in our preliminary work at NeurIPS'2022 [1], which we call Iso-Dream, the learning process of inverse dynamics is prone to the problem of "training collapse", where the action-conditioned branch captures all dynamic information, while the action-free branch learns almost nothing. To further isolate different dynamics in an unsupervised manner, we use new forms of min-max variance constraints to regularize the information flow of dynamics in the decoupled world model. More concretely, we provide a batch of hypothetical actions to the world

- *The authors are with MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China.*
- *Corresponding author: Y. Wang, yunbow@sjtu.edu.cn.*
- *Code: https://github.com/panmt/MBRL_with_Isolated_Imaginations.*

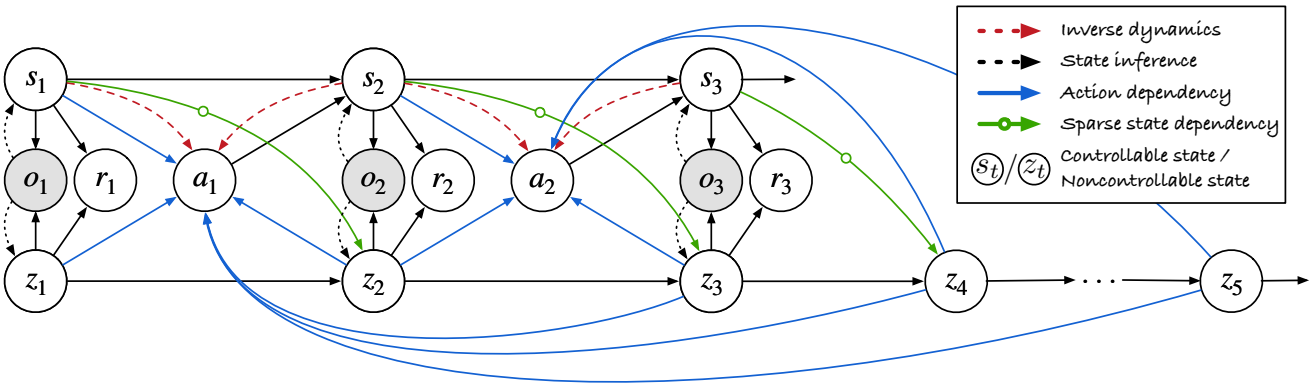


Fig. 1: Graphic model of our approach. The world model learns to decouple mixed visual dynamics into controllable states (s_t) and noncontrollable states (z_t) by optimizing the inverse dynamics (as indicated by the red dashed arrows). With state decoupling, the RL agent can make decisions based on the forecasts of future noncontrollable dynamics of the environment (blue arrows). In forward modeling, we consider the sparse dependency of next-step noncontrollable states on current controllable states (green arrows). In representation learning, we further cope with the imbalance of dynamic information learned in different state transition branches.

model, and encourage the action-conditioned branch to produce different state transitions based on the same state, while penalizing the diversity of those in the action-free branch.

1.2 How to improve behavior learning based on decoupled world models?

Humans can decide how to interact with the environment at each moment based on their anticipation of future changes in their surroundings. Accordingly, by decoupling the state transitions, our approach can explicitly forecast the evolution of action-independent dynamics in the system, thereby greatly benefiting downstream decision-making tasks. Unlike Dreamer, it performs latent state imagination in both the training phase and testing phase of the agent behaviors to make more forward-looking decisions. As shown by the blue arrows in Fig. 1, the policy network integrates the current controllable state and multiple steps of predicted noncontrollable states through an attention mechanism. Intuitively, since future noncontrollable states at different steps may have different weights of impact on the current decision of the agent, the attention mechanism enables the agent to adaptively consider possible future interactions with the environment. It ensures that only appropriate future states are fed back into the policy.

Despite the effectiveness of the new behavior learning scheme, it only considers the indirect influence of action-free dynamics on future action-conditioned dynamics through agent behaviors (*i.e.*, $z_{t:t+\tau} \rightarrow a_t \rightarrow s_{t+1}$). Another improvement of our approach over Iso-Dream is that it further models the *sparse dependency* of future noncontrollable states on current controllable states (*i.e.*, $s_t \rightarrow z_{t+1}$), which is indicated by the green arrows in Fig. 1. In practical scenarios, for example, when we program a robot to compete with another one in a dynamic game, the opponent can adjust its policy according to the behavior of our agent. In autonomous driving, when an agent vehicle veers into the lane of other vehicles, typically those vehicles will slow down to avoid a collision. Because of the proposed solution to training collapse, modeling the sparse dependency does not affect the disentanglement learning ability of the world model. In behavior learning, actions are sampled from $\pi(s_t, z_{t:t+\tau})$, where $z_{t+1} \sim p(\cdot|z_t, s_t)$, while due to the sparsity of the cross-branch dependencies, the long-horizon noncontrollable future can be approximated as $z_{t+2:t+\tau} \sim p(\cdot|z_{t+1:t+\tau-1})$.

We evaluate Iso-Dream++ in the following domains: (i) the CARLA autonomous driving environment in which other vehicles can be naturally viewed as noncontrollable components; (ii) the modified DeepMind Control Suite with noisy video background. Our approach outperforms existing approaches by large margins and further achieves significant advantages in transfer learning by isolating dynamics. It can selectively transfer controllable or noncontrollable parts of the learned state transition functions from the source domain to the target domain according to the prior information.

The main contributions of this paper are summarized as follows:

- We present a new world model and encourage the decomposition of latent state transitions by optimizing *inverse dynamics*.
- We introduce the *min-max variance constraints* to prevent all information from collapsing into a single state transition branch.
- We improve the actor-critic algorithm to make *forward-looking decisions* based on the forecasts of future noncontrollable states.
- We model the *sparse dependency* of the next-step noncontrollable dynamics on current controllable dynamics to provide a more accurate simulation of some practical dynamic environments.
- We empirically demonstrate the advantages of Iso-Dream++ over existing methods in standard, noisy, and *transfer learning* setups.

In summary, we extend our previous studies with (i) the min-max variance constraints, (ii) the sparse dependence between the decoupled latent states, and (iii) the transfer learning experiments.

2 Problem Overview

2.1 Problem Definition

In visual control tasks, the agent learns the action policy directly from high-dimensional observations. We formulate visual control as a partially observable Markov decision process (POMDP) with a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{O}$ is the observation space, $\mathcal{R}(s_t, a_t)$ is the reward function, and $\mathcal{T}(s_{t+1} \mid s_t, a_t)$ is the state-transition distribution. At each timestep $t \in [1; T]$, the agent takes an action $a_t \in A$ to interact with the environment and receives a reward $r_t = \mathcal{R}(s_t, a_t)$. The objective is to learn a policy that maximizes the expected cumulative reward $\mathbb{E}_p[\sum_{\tau=1}^{T} r_\tau]$. In this setting, the agent cannot access the true states in $\mathcal{S}$.

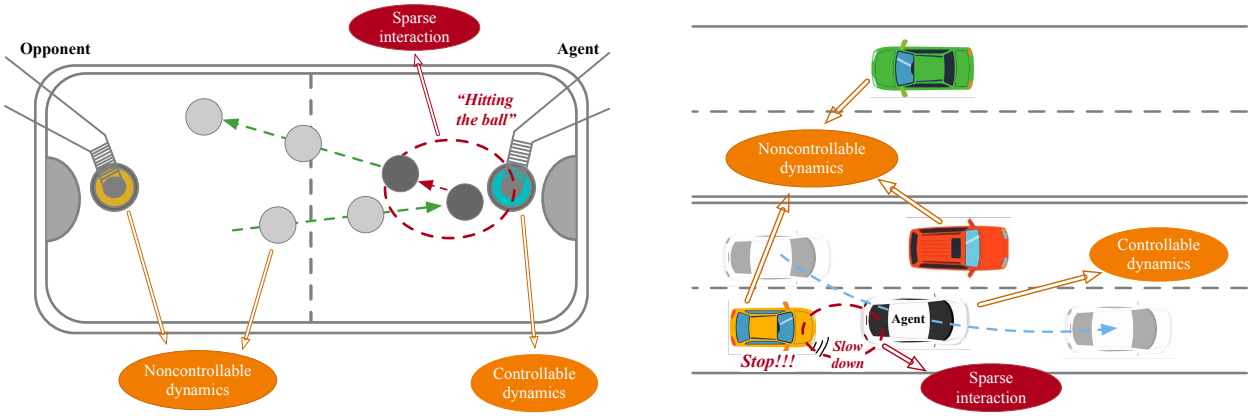


Fig. 2: Examples of sparse dependency of the noncontrollable state on the controllable state. (a) A game resembling ice hockey that is played on a desk, in which the ego-agent is controllable, while the opponent robot can be seen as the noncontrollable part, and the hockey puck can be mostly considered to have predictable dynamics independent of the agent's actions. *Sparse dependency* occurs at the moment the agent hits the puck because it may change direction depending on the agent's current state. (b) In autonomous driving, other vehicles (*yellow*) will change their driving directions to avoid collision when the ego-agent (*white*) takes up their driveway.

2.2 Key Challenges

Challenge 1: How to learn future-conditioned policies without the expensive Monte-Carlo planning? Forecasting future environmental changes is useful for decision-making in a non-stationary system. A typical solution, such as the cross-entropy method (CEM), is to perform Monte-Carlo sampling over future actions and value the consequences of multiple action trajectories [10, 4, 11]. These algorithms are expensive in computational cost, especially when we have large action and state spaces. The question is: *Can we design an RL algorithm that allows for future-conditioned decision-making without playing dice in the action space?*

Challenge 2: How to avoid "training collapse" in unsupervised dynamics disentanglement? Despite the great success in unsupervised representation learning [12, 13, 14], it remains a challenge to disentangle the controllable and noncontrollable dynamic patterns in non-stationary visual scenes. One potential solution is to employ modular structures that learn different dynamics in separate branches. However, without proper constraints, the model may suffer from "training collapse", where one branch captures all useful information and the other learns almost nothing. This phenomenon may occur when the noncontrollable dynamics components are easy to predict. In this case, we consider adding further constraints to the learning objects of the action-conditioned and action-free state transition branches, encouraging them to isolate the noncontrollable part from the mixed dynamics.

Challenge 3: How to model situations where the agent behavior has only a sparse/indirect impact on noncontrollable dynamics? As we know, in realistic scenarios, the noncontrollable component of the dynamics may not evolve independently but may depend on the motions of the controllable component. For instance, in Fig. 2 (a), the hockey puck on the desk (noncontrollable part) changes its direction when the agent (controllable part) hits it. For autonomous driving, in Fig. 2 (b), other vehicles (noncontrollable part) will slow down to avoid a collision when the agent (controllable part) takes their lane. If we assume that our actions do not indirectly affect other vehicles on the road, then for safety reasons, a sub-optimal policy for handling heavy traffic could be to follow the vehicle in front of us instead of changing lanes. Accordingly, we propose a sparse dependency mechanism that enhances our model's decision-making ability. Empirical results are illustrated in Fig. 9.

2.3 Basic Assumptions

In our proposed framework as shown in Fig. 1, when the agent receives a sequence of visual observations $o_{1:T}$, the underlying spatiotemporal dynamics can be defined as $u_{1:T}$. The evolution of different dynamics can be caused by different forces, but here we aim to decouple $u_{1:T}$ into controllable latent states $s_{1:T}$ and time-varying noncontrollable latent states $z_{1:T}$, such that:

$$\begin{aligned} u_{1:T} &\sim (s, z)_{1:T}, \\ s_{t+1} &\sim p(s_{t+1} \mid s_t, a_t), \\ z_{t+1} &\sim p(z_{t+1} \mid z_t), \end{aligned} \tag{1}$$

where a_t is the action signal. By isolating s_t and z_t to each other, we model their state transitions of $p(s_{t+1} \mid s_t, a_t)$ and $p(z_{t+1} \mid z_t)$ respectively. We assume that a more clear decoupling of s_t and z_t can benefit both long-term predictions and decision-making. As an extension of our preliminary work [1], we additionally model the sparse dependency of noncontrollable dynamics on controllable dynamics (as described below). Thus, when a sparse event is detected, the transition of noncontrollable state in Eq. (1) can be rewritten as $z_{t+1} \sim p(z_{t+1} \mid z_t, s_t)$.

It assumes that the agent can greatly benefit from predicting the consequences of external noncontrollable forces. During behavior learning, we roll out the noncontrollable states and then associate them with the current controllable states for more proactive decision-making. We derive the action policy by

$$a_t \sim \pi(a_t \mid s_t, \mathbb{1} \odot z_{t:t+\tau}), \tag{2}$$

where $\mathbb{1}$ is an indicator according to our prior knowledge about the environment. For example, in autonomous driving, since it is reasonable for the ego-agent to make decisions based on the predictions about the future states of other vehicles, we have $\mathbb{1} = 1$ and calculate the relations between s_t and the imagined noncontrollable states in a time horizon τ. Otherwise, for some specific tasks where the noncontrollable components are irrelevant to decision-making, we can simply set the indicator function to $\mathbb{1} = 0$ and treat them as noisy distractions.

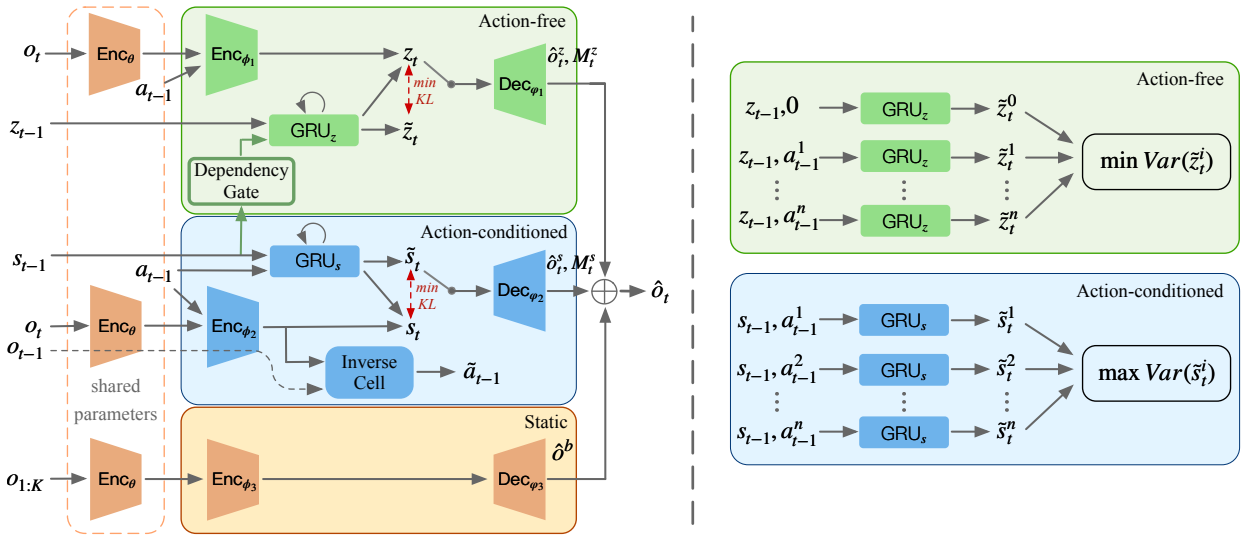


Fig. 3: The overall architecture of the world model in Iso-Dream++. **Left:** The world model has three branches to explicitly disentangle controllable and noncontrollable state transitions, as well as the static components from visual data. **Right:** Illustration of calculating variance in different branches. Given the different action signals, our objective is to minimize the diversity of state transitions in the action-free branch and maximize the diversity of those in the action-conditioned branch.

3 Method

In this section, we present the technical details of Iso-Dream++ for decoupling and leveraging controllable and noncontrollable dynamics for visual MBRL. The overall pipeline is based on Dreamer [6], where we learn the world model from a dataset of past experience, learn behaviors from imagined sequences of compact model states, and execute the behavior policy in the environment to grow the experience dataset.

In Section 3.1.1, we first introduce the three-branch world model and its training objectives of *inverse dynamics*. In Section 3.1.2, we propose the *min-max variance constraints* to regularize the dynamics representations in each state transition branch to enhance disentanglement learning and avoid training collapse. In Section 3.1.3, we present a network structure to model the phenomenon that future noncontrollable dynamics can be sparsely influenced by current controllable dynamics. In Section 3.2, we present an actor-critic method that is trained on the imaginations of the decoupled world model latent states, so that the agent may consider possible future states of noncontrollable dynamics in behavior learning. Finally, in Section 3.3, we discuss how our model is deployed to interact with the environment.

3.1 World Models with Dynamics Isolation

Inspired by prior research [15, 16] that demonstrates the efficacy of modular structures for disentanglement learning, we use an architecture with multiple branches to model different dynamics independently, according to their respective physical laws. Each individual branch tends to present robust features, even when the dynamic patterns in other branches undergo changes. Specifically, our three-branch model, illustrated in the left panel of Fig. 3, disentangles visual observations into controllable dynamics state s_t, noncontrollable dynamics state z_t, and a time-invariant component of the environment. The action-conditioned branch models the controllable state transition $p(s_{t+1} \mid s_t, a_t)$. It follows the RSSM architecture from PlaNet [11] to use a recurrent neural network $\texttt{GRU}_s(\cdot)$, the deterministic hidden state h_t, and the stochastic state s_t to form the transition model, where the GRU keeps the historical information of the controllable dynamics. The action-free branch models $p(z_{t+1} \mid z_t)$ with similar network structures. The transition models with separate parameters can be written as follows:

$$\begin{aligned} p(\tilde{s}_t \mid s_{<t}, a_{<t}) &= p(\tilde{s}_t \mid h_t), \\ p(\tilde{z}_t \mid z_{<t}) &= p(\tilde{z}_t \mid h'_t), \end{aligned} \tag{3}$$

where $h_t = \texttt{GRU}_s(h_{t-1}, s_{t-1}, a_{t-1})$, $h'_t = \texttt{GRU}_z(h'_{t-1}, z_{t-1})$. We here use $\tilde{s}_t$ and $\tilde{z}_t$ to denote the prior representations. We optimize the transition models with posterior representations that are derived from $s_t \sim q(s_t \mid h_t, o_t, a_{t-1})$ and $z_t \sim q(z_t \mid h'_t, o_t, a_{t-1})$. We learn the posteriors from the observation at current time step $o_t \in \mathbb{R}^{3\times H\times W}$ by a shared encoder $\texttt{Enc}_\theta$ and subsequent branch-specific encoders $\texttt{Enc}_{\phi_1}$ and $\texttt{Enc}_{\phi_2}$. Notably, we feed actions into both $\texttt{Enc}_{\phi_1}$ and $\texttt{Enc}_{\phi_2}$, which differs from our previous work [1]. In the static branch, where there is no state transition, we only use an encoder $\texttt{Enc}_{\phi_3}$ and a decoder $\texttt{Dec}_{\varphi_3}$ to model simple time-invariant information in the environment.

3.1.1 Inverse Dynamics

To enable disentanglement representation learning that corresponds to the control signals, we introduce the training objective of *inverse dynamics*. This objective encourages the action-conditioned branch to learn a more deterministic state transition based on specific actions, while the action-free branch learns the remaining noncontrollable dynamics independent of the control signals. Accordingly, we design an *Inverse Cell* of a 2-layer MLP to infer the actions that lead to certain transitions of the controllable states:

$$\text{Inverse dynamics:} \quad \tilde{a}_{t-1} = \texttt{MLP}(s_{t-1}, s_t), \tag{4}$$

where the inputs are the posterior representations in the action-conditioned branch. By learning to regress the true behavior a_{t-1}, the Inverse Cell facilitates the action-conditioned branch to isolate the representation of the controllable dynamics. We respectively use the prior state $\tilde{s}_t$ and the posterior state z_t to generate the controllable visual component $\hat{o}_t^s \in \mathbb{R}^{3\times H\times W}$ with mask $M_t^s \in \mathbb{R}^{1\times H\times W}$ and the noncontrollable component $\hat{o}_t^z \in \mathbb{R}^{3\times H\times W}$

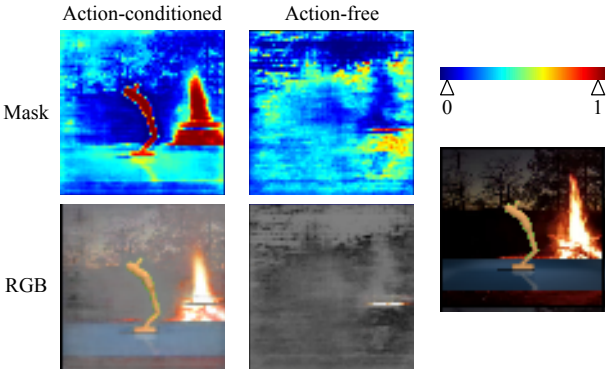


Fig. 4: A showcase of training collapse that the action-conditioned branch in the original version of Iso-Dream dominates the learning process of both controllable and noncontrollable dynamics. The corresponding results of Iso-Dream++ are shown in Fig. 14.

with $M_t^z \in \mathbb{R}^{1\times H\times W}$. By further integrating the static information extracted from the first K frames, we have

$$\hat{o}_t = M_t^s \odot \hat{o}_t^s + M_t^z \odot \hat{o}_t^z + (1 - M_t^s - M_t^z) \odot \hat{o}^b, \quad (5)$$

where $\hat{o}^b = \texttt{Dec}_{\varphi_3}(\texttt{Enc}_{\theta,\phi_3}(o_{1:K}))$.

For reward modeling, we have two options concerning the action-free branch. First, we may regard noncontrollable dynamics as irrelevant noises that do not contribute to the task and therefore do not involve z_t in imagination. In other words, the policy and predicted rewards would solely rely on controllable states, *e.g.*, $p(r_t \mid s_t)$. Alternatively in other cases, we need to consider the influence of future noncontrollable states on the agent's decision-making process and incorporate the action-free components during behavior learning. To achieve this, we train the reward predictor to model $p(r_t \mid s_t, z_t)$ in the form of MLPs.

For a training sequence of $(o_t, a_t, r_t)_{t=1}^T$ sampled from the replay buffer, the world model can be optimized using the following loss functions, where α, β_1, and β_2 are hyper-parameters:

$$\begin{aligned}\mathcal{L}_{\text{base}} =& \mathbb{E}\{\sum_{t=1}^{T} \underbrace{-\ln p(o_t \mid h_t, s_t, h'_t, z_t)}_{\text{image log loss}} + \underbrace{\alpha \ell_2(a_t, \tilde{a}_t)}_{\text{action loss}} \\ & \underbrace{-\ln p(r_t \mid h_t, s_t, h'_t, z_t)}_{\text{reward log loss}} \underbrace{-\ln p(\gamma_t \mid h_t, s_t, h'_t, z_t)}_{\text{discount log loss}} \\ & + \underbrace{\beta_1 \text{KL}[q(s_t \mid h_t, o_t) \mid p(s_t \mid h_t)]}_{\text{KL divergence in the action-conditioned branch}} \\ & + \underbrace{\beta_2 \text{KL}[q(z_t \mid h'_t, o_t) \mid p(z_t \mid h'_t)]}_{\text{KL divergence in the action-free branch}}\}.\end{aligned} \quad (6)$$

3.1.2 Training Collapse and Min-Max Variance Constraints

Despite modeling inverse dynamics, for the original Iso-Dream, we find that it is still challenging for the world model to isolate controllable and noncontrollable dynamics. We observed in the preliminary experiments that the disentanglement results were unstable over multiple runs of world model training, and that the action-conditioned branch occasionally learned mismatched representations of noncontrollable state transitions. An example is shown in Fig. 4. It implies that most useful information may collapse into the action-conditioned branch while the action-free branch learns almost nothing, which we call "*training collapse*". This phenomenon arises due to the inherent limitations of the training objective in inverse dynamics, which may not always ensure the complete exclusion of action-independent state transitions, particularly when the action-conditioned network branch possesses a strong capacity for modeling dynamics.

To keep the state transition branches from training collapse, we propose the *min-max variance constraints*, whose key idea is to (i) maximize the diversity of outcomes in the action-conditioned branch given distinct action inputs and (ii) minimize the diversity of outcomes in the action-free branch under similar conditions. To this end, unlike in the original Iso-Dream, we make the action-free branch also aware of the action signal during the world model learning process. But for behavior learning and policy deployment, we simply set the input action to 0-values.

There is an information-theoretic interpretation behind calculating variance. In order to investigate the connection between dynamics and action signals, the world model is enforced to identify the dynamics that provide information about our beliefs regarding the action signals. The expected information gain can be expressed as the conditional entropy of state and action:

$$I(s_t; a_{t-1} | s_{t-1}) = H(s_t | s_{t-1}) - H(s_t | s_{t-1}, a_{t-1}). \quad (7)$$

As shown in Fig. 3 (right), for the action-conditioned branch, we maximize the mutual information between the state and action signal to focus on the state transition of a specific action. Given a batch of hypothetical actions $\{a_{t-1}^i | i \in [1, n]\}$, for the same controllable state s_{t-1}, we have different state transitions based on these actions: $\tilde{s}_t^i \sim p(\tilde{s}_t^i | s_{t-1}, a_{t-1}^i),\ i \in [1, n]$. The empirical variance is used to approximate the information gain, and the objective can be written as

$$\begin{aligned} L_s &= \max \sum_t^T \text{Var}(\tilde{s}_t^i) = \max \sum_t^T \frac{1}{n-1} \sum_i (\tilde{s}_t^i - \bar{s}_t)^2, \\ \bar{s}_t &= \frac{1}{n} \sum_i \tilde{s}_t^i, \quad i \in [1, n]. \end{aligned} \quad (8)$$

On the contrary, in the action-free branch, we minimize the variance of output states resulting from different actions, penalizing the diversity of state transitions:

$$\begin{aligned} L_z &= \min \sum_t^T \text{Var}(\tilde{z}_t^i) = \min \sum_t^T \frac{1}{n-1} \sum_i (\tilde{z}_t^i - \bar{z}_t)^2, \\ \bar{z}_t &= \frac{1}{n} \sum_i \tilde{z}_t^i, \quad i \in [1, n]. \end{aligned} \quad (9)$$

The overall training objective of the world model is

$$L_{\text{all}} = L_{\text{base}} + L_{\text{var}}, \quad (10)$$

where $L_{\text{var}} = \lambda_1 L_s + \lambda_2 L_z$. λ_1 and λ_2 are hyper-parameters.

For convenience, we only use two opposite actions $\{a_t, -a_t\}$ in the action-conditioned branch, and use the action set $\{a_t, 0, -a_t\}$ in the action-free branch to figure out L_s and L_z. As for subsequent learning, we use a_t and 0 in the action-conditioned and action-free branches, respectively.

3.1.3 Sparse Dependency between Decoupled States

In certain situations, the controllable and noncontrollable dynamics are not entirely independent, as shown in Fig. 2. This is particularly true in autonomous driving, where the actions of the ego-agent can influence the behavior of other vehicles, causing them to steer or slow down. To accurately predict future noncontrollable states

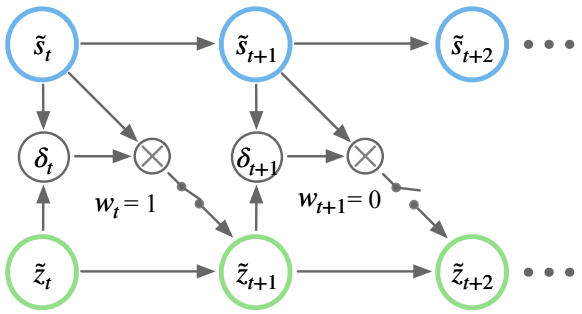


Fig. 5: The dependency gate involves a binary gate that can either be open ($w_t = 1$) or closed ($w_t = 0$). When the gate is open, the transition of the next noncontrollable state $\tilde{z}_{t+1}$ takes into account the dependency between $\tilde{s}_t$ and $\tilde{z}_t$.

based on current controllable states, it is essential to account for these sparse dependencies.

To achieve effective modeling of sparse dependency, it is essential to identify the moment when controllable states exert a significant influence on noncontrollable states. In order to facilitate this, we present a compact module called the *dependency gate*, which connects the previously isolated action-free and action-conditioned branches, as shown in Fig. 3 (left). We unfold the detailed structure dependency gate in time (see Fig. 5), where the controllable state $\tilde{s}_t$ and noncontrollable state $\tilde{z}_t$ are concatenated and passed through a fully connected layer represented by $f(\tilde{s}_t, \tilde{z}_t)$. A sigmoid function is then applied as an activation signal to control the gate, which is formulated as

$$\delta_t(w_t = 1 | \tilde{s}_t, \tilde{z}_t) = \begin{cases} 1, & \text{sigmoid}(f(\tilde{s}_t, \tilde{z}_t)) \geq 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

When the gate detects a dependency between controllable and noncontrollable states ($w_t = 1$), the subsequent noncontrollable state $\tilde{z}_{t+1}$ is determined by both $\tilde{s}_t$ and $\tilde{z}_t$ using the action-free transition, which is defined as follows:

$$\tilde{z}_{t+1} \sim p(\tilde{z}_{t+1} \mid \tilde{z}_t, w_t \odot \tilde{s}_t). \tag{12}$$

3.2 Behavior Learning in Isolated Imaginations

Thanks to the decoupled world model, we can optimize agent behavior to adaptively consider the relationship between available actions and potential future states of the noncontrollable dynamics. A practical example is autonomous driving, where the movement of other vehicles can be naturally viewed as noncontrollable but predictable components. As shown in Fig. 6, we here propose an improved actor-critic learning algorithm that *(i) allows the action-free branch to foresee the future ahead of the action-conditioned branch, and (ii) exploits the predicted future information of noncontrollable dynamics to make more forward-looking decisions*.

Suppose we are making decisions at time step t in the imagination period. A straightforward solution from the original Dreamer method is to learn an action model and a value model based on the isolated controllable state $\tilde{s}_t \in \mathbb{R}^{1\times d}$. With the aid of an attention mechanism, we can establish a connection between it and future noncontrollable states. It is important to note that we only employ sparse dependency in the initial imagination step to obtain $\tilde{z}_{t+1}$, as the subsequent controllable states are not yet available at time step t. Once we have predicted a sequence of future noncontrollable states $\tilde{z}_{t:t+\tau} \in \mathbb{R}^{\tau\times d}$, where τ is the sliding

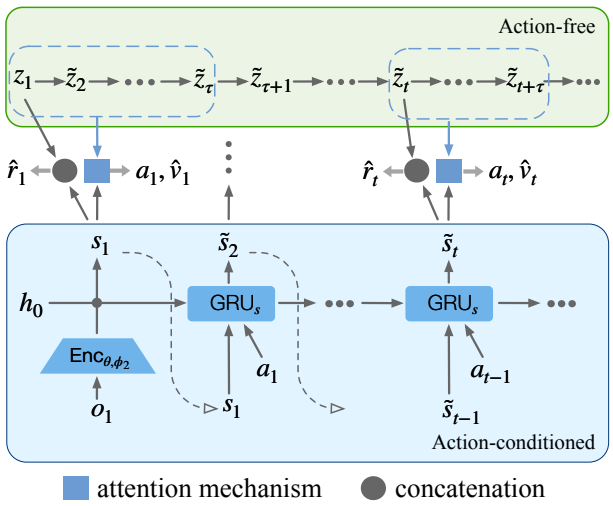


Fig. 6: The agent learns *future-dependent* policies in world model's imaginations through a *future state attention* mechanism.

window length from the present time, we explicitly compute the relations between them using the following equation:

$$e_t = \text{softmax}(\tilde{s}_t \, \tilde{z}_{t:t+\tau}^T) \, \tilde{z}_{t:t+\tau} + \tilde{s}_t. \tag{13}$$

This equation allows us to dynamically adjust the horizon of future noncontrollable states using the attention mechanism. In this way, $\tilde{s}_t$ evolves to a more "*visionary*" representation $e_t \in \mathbb{R}^{1\times d}$. We modify the action and value models in Dreamer [6] as follows:

$$\begin{aligned} &\text{Action model:} \quad a_t \sim \pi(a_t \mid e_t), \\ &\text{Value model:} \quad v_\xi(e_t) \approx \mathbb{E}_{\pi(\cdot|e_t)} \sum_{k=t}^{t+L} \gamma^{k-t} r_k, \end{aligned} \tag{14}$$

where L is the imagination time horizon. As shown in Alg. 1, during imagination, we first use the action-free transition model to obtain sequences of noncontrollable states of length $L+\tau$, denoted by $\{\tilde{z}_i\}_{i=t}^{i+L+\tau}$. At each time step in the imagination period, the agent draws an action a_j from the visionary state e_j, which is derived from Eq. (13). The action-conditioned branch uses the action a_j in latent imagination and predicts the next controllable state s_{j+1}. We follow DreamerV2 [8] to train our action model with the objective of maximizing the λ-return [17], while our value model was trained to perform regression on the λ-return. For further information on the loss functions, please refer to Eq. (5-6) as detailed in the paper of DreamerV2 [8].

3.3 Policy Deployment by Rolling out Noncontrollable Dynamics

During policy deployment, as shown in Lines 22-24 in Alg. 1, the action-free branch predicts the next-step noncontrollable states $\tilde{z}_{t+1}$ using Eq. (12) and then consecutively rolls out the future noncontrollable states $\tilde{z}_{t+2:t+\tau}$ starting from $\tilde{z}_{t+1}$. Similar to Eq. (13) used in the process of behavior learning, the learned future state attention network is used to adaptively integrate s_t, z_t and $\tilde{z}_{t+1:t+\tau}$. Based on the integrated feature e_t, the Iso-Dream++ agent then draws a_t from the action model to interact with the environment. As discussed in Section 2.2, if the noncontrollable dynamics are irrelevant to the control task, the policy at each time step t is generated using only the state of controllable dynamics when interacting with the environment.

Algorithm 1 Iso-Dream++ (Highlight: Our modifications to behavior learning & policy deployment of the original Dreamer)

Hyper-parameters: L: Imagination horizon; τ: Window size for future state attention

Initialize the replay buffer $\mathcal{B}$ with random episodes.
while not converged **do**
  for update step $c = 1 \dots C$ **do**
    `// Representation learning`
    Draw data sequences $\{(o_t, a_t, r_t)\}_{t=1}^{T} \sim \mathcal{B}$.
    Compute the controllable state $s_t \sim q(s_t|h_t, o_t, a_{t-1})$ and the noncontrollable state $z_t \sim q(z_t|h'_t, o_t, a_{t-1})$.
    Compute world model loss using Eq. (10) and update model parameters.
    `// Behavior learning`
    for time step $i = t \dots t + L$ **do**
      Compute the next noncontrollable state $\tilde{z}_{i+1}$ using Eq. (12).
      Roll-out the noncontrollable states $\{\tilde{z}_j\}_{j=i+2}^{i+\tau}$ from $\tilde{z}_{i+1}$ through the action-free branch alone.
      Compute latent state $e_i \sim \texttt{Attention}(\tilde{s}_i, \tilde{z}_{i:i+\tau})$ using Eq. (13).
      Imagine an action $a_i \sim \pi(a_i|e_i)$.
      Predict the next controllable state $\tilde{s}_{i+1} \sim p(\tilde{s}_i, a_i)$ using the action-conditioned branch alone.
    end for
    Update the policy and value models in Eq. (14) using estimated rewards and values.
  end for
  `// Environment interaction`
  $o_1 \leftarrow \texttt{env.reset}()$
  for time step $t = 1 \dots T$ **do**
    Calculate the posterior representation $s_t \sim q\left(s_t \mid h_t, o_t, a_{t-1}\right), z_t \sim q\left(z_t \mid h'_t, o_t, a_{t-1}\right)$.
    Compute the next noncontrollable state $\tilde{z}_{t+1} \sim p(\tilde{z}_{t+1}|z_t, w_t \odot s_t)$ using Eq. (12).
    Roll-out the noncontrollable states $\tilde{z}_{t+2:t+\tau}$ from $\tilde{z}_{t+1}$ through the action-free branch alone.
    Generate $a_t \sim \pi(a_t \mid s_t, z_t, \tilde{z}_{t+1:t+\tau})$ using future state attention in Eq. (13).
    $r_t, o_{t+1} \leftarrow \texttt{env.step}(a_t)$
  end for
  Add experience to the replay buffer $\mathcal{B} \leftarrow \mathcal{B} \cup \{(o_t, a_t, r_t)_{t=1}^{T}\}$.
end while

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate Iso-Dream++ on two RL environments:

- **CARLA** [18]: CARLA is a simulator with complex and realistic visual observations for autonomous driving research. We train our model to perform the task of first-person highway driving in "Town04", where the agent's goal is to drive as far as possible in 1,000 time steps without colliding with any of the 30 other moving vehicles or barriers. In addition to our conference paper, we incorporate more diverse settings into our study, including both day and night modes as shown in Fig. 7.
- **DeepMind Control Suite** [19]: DMC contains a set of continuous control tasks and serves as a standard benchmark for vision-based RL. To evaluate the generalization of our method by disentangling different components under complex visual dynamics, we use two modified benchmarks [20], namely `video_easy`, which contains 10 simple videos, and `video_hard`, which contains 100 complex videos.

Compared methods. We compare Iso-Dream++ with the following visual RL approaches:

- **DreamerV2 [8]**: A model-based RL method that learns directly from latent variables in world models. The latent representation allows agents to imagine thousands of trajectories in parallel.
- **DreamerV3 [21]**: A further improved version of Dreamer that learns to master diverse domains with fixed hyperparameters.
- **DreamerPro [22]**: A non-contrastive, reconstruction-free model-based RL method that combines Dreamer [6] with prototypes to enhance robustness to distractions.
- **CURL [23]**: A model-free RL method that uses contrastive learning to extract high-level features from raw pixels, maximizing agreement between augmented data of the same observation.

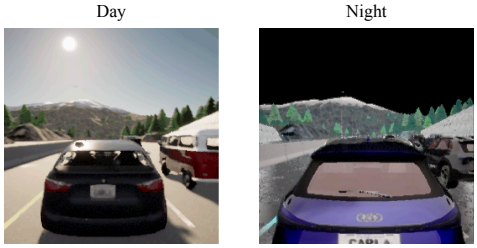


Fig. 7: Examples of day and night modes in CARLA.

- **SVEA [24]**: A framework for data augmentation in deep Q-learning algorithms that improves stability and generalization on off-policy RL.
- **SAC [25]**: A model-free actor-critic method that optimizes a stochastic policy in an off-policy way.
- **DBC [26]**: A method that learns a bisimulation metric representation without reconstruction loss. This representation is invariant to different task-irrelevant details in the observation.
- **Denoised-MDP [27]**: A framework that categorizes information out in the wild into four types based on controllability and relation with reward, and formulates useful information as that which is both controllable and reward-relevant.

4.2 CARLA Autonomous Driving Environment

Implementation. In the autonomous driving task, We use a camera with a 60 degree view on the roof of the ego-vehicle, which obtains images of 64×64 pixels. Following the setting in the DBC [26], in order to encourage highway progression and penalize collisions, the reward is formulated as $r_t = v_{ego}^T \hat{u}_h \cdot \Delta t - \xi_1 \cdot \mathbb{1} - \xi_2 \cdot |steer|$, where v_{ego} is the velocity vector of the ego-vehicle, projected onto the highway's unit vector $\hat{u}_h$, and multiplied by time discretization $\Delta t = 0.05$ to measure highway progression in meters. We use

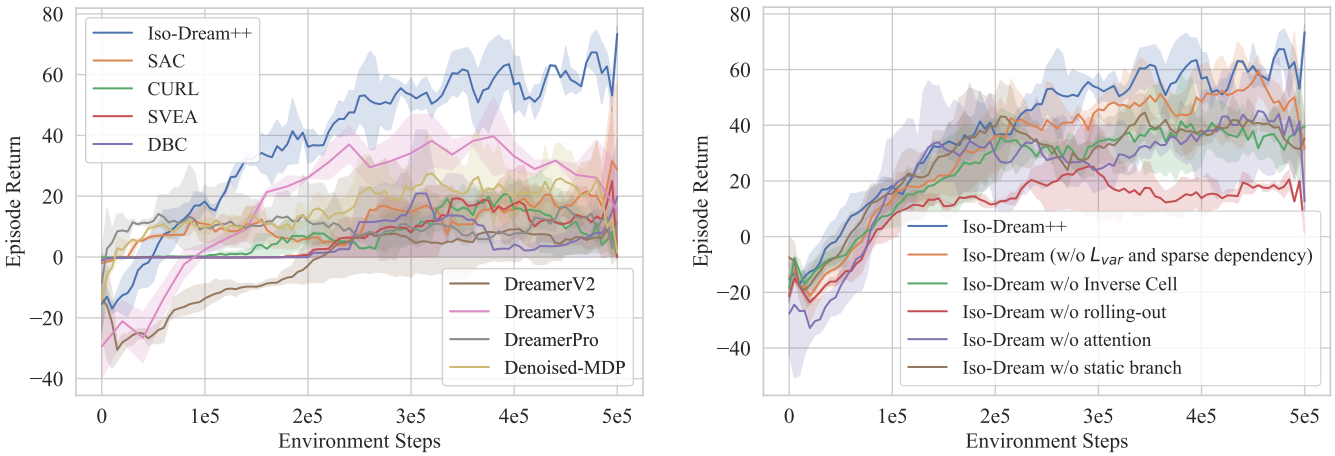


Fig. 8: (a) Quantitative comparison with existing approaches for CARLA driving. (b) Ablation studies of individual effectiveness of inverse dynamics optimization (green), noncontrollable rollouts (red), future-state attention (purple), and the separate branch for static information modeling (brown). We also compare Iso-Dream++ with its predecessor in our conference paper (orange).

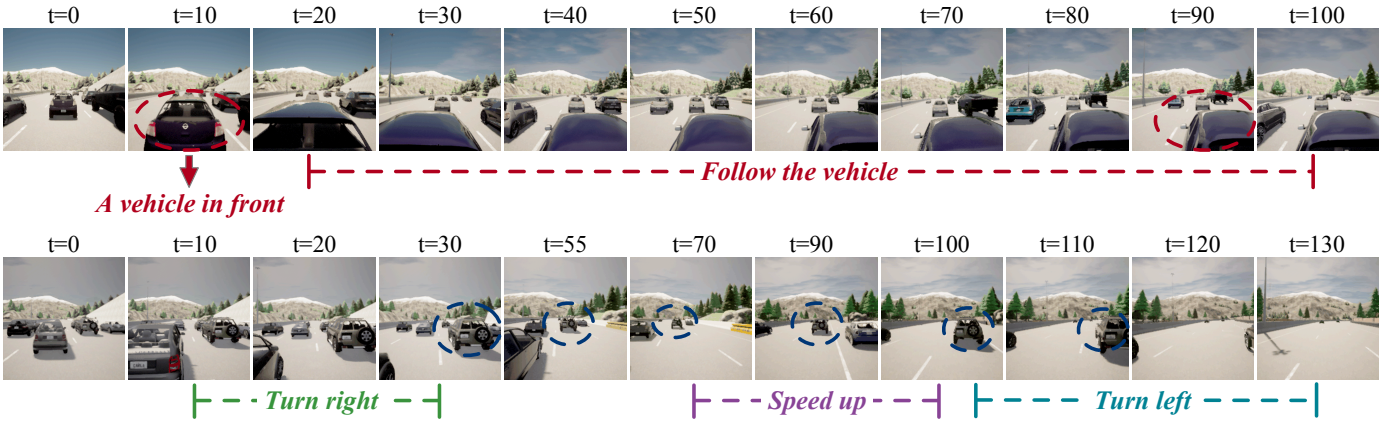


Fig. 9: Examples of Iso-Dream++ without (**top**) and with (**bottom**) sparse dependency. The agent without sparse dependency tends to follow the vehicle in front of it when there are many vehicles in the way. In the bottom row, the agent overtakes and accelerates flexibly.

$\mathbb{1} \in \mathbb{R}^+$ for collisions and a steering penalty $steer \in [-1, 1]$ to facilitate lane-keeping. The hyper-parameters are set to $\xi_1 = 10^{-4}$ and $\xi_2 = 1$, respectively. We use $\beta_1 = \beta_2 = 1$ and $\alpha = 1$ in Eq. (6), $\lambda_1 = \lambda_2 = 1$ in Eq. (10), and $\tau = 5$ in Eq. (13).

Quantitative comparisons. We present the quantitative results in CARLA in Fig. 8(a). Iso-Dream++ outperforms the compared models, including DreamerV2, DreamerV3, DreamerPro, and Denoised-MDP, significantly. After $500k$ environment steps, Iso-Dream++ achieves an average return of around 60, while DreamerV2 and Denoised-MDP achieve 10 and 25 respectively. In DreamerV2, the latent representations contain both controllable and noncontrollable dynamics, which increases the complexity of modeling the state transitions in imagination. Compared with Denoised-MDP, which also decouples information according to controllability, Iso-Dream++ has the advantage of making forward-looking decisions by rolling out future noncontrollable states.

Ablation studies. Fig. 8(b) provides the ablation study results that validate the effectiveness of inverse dynamics, the rolling-out strategy of noncontrollable states, the attention mechanism, and the modeling of static information. As shown by the green curve, removing the Inverse Cell reduces the performance of Iso-Dream++, which indicates the importance of isolating controllable and noncontrollable components. For the model indicated by the red curve, we do not use the rollouts of future noncontrollable states as the inputs of the action model. The results demonstrate that rolling out noncontrollable states in the action-free branch significantly improves the agent's decision-making results by perceiving potential risks in advance. Moreover, we evaluate Iso-Dream++ without attention mechanism where the action model directly concatenates the current controllable state with a sequence of future noncontrollable states and takes them as inputs. As shown by the purple curve, the attention mechanism extracts valuable information from future noncontrollable dynamics better than concatenation. Furthermore, as shown by the brown curve, our approach's performance decreases by about 15% without a separate network branch for capturing the static information. Moreover, a comparison between the blue curve and the orange curve reveals a decline in our model's performance when we remove the min-max variance constraints and sparse dependency modeling. Unlike the DMC suite, where the original Iso-Dream is more vulnerable to training collapse, in the CARLA environment, the sparse dependency modeling method plays a crucial role in the improved performance of Iso-Dream++. In Fig. 9, we present

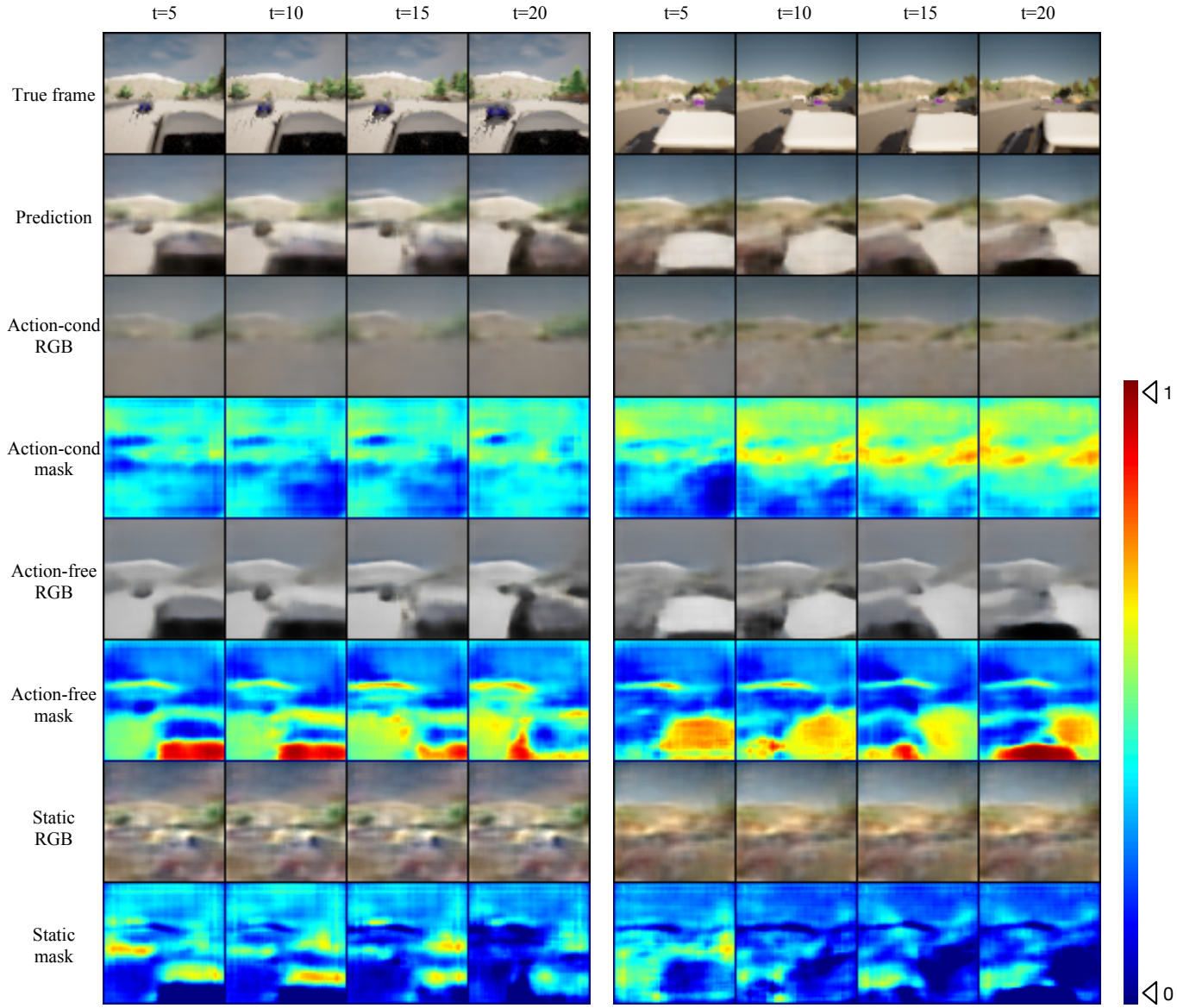


Fig. 10: Video prediction results in CARLA. For each sequence, we use the first 5 images as context frames. The visual decoupled components (Rows 3, 5, 7) and masks (Rows 4, 6, 8) of each branch are presented. Iso-Dream++ successfully isolates noncontrollable dynamics from the complicated environment, *i.e.*, other driving vehicles.

visual examples produced by our models with and without the sparse dependency. Without sparse dependency (top row), the agent fails to predict that other vehicles will slow down or brake when changing lanes, making it safer to follow the vehicle ahead rather than overtake it during traffic congestion. However, as shown in the bottom row of Fig. 9, the agent can decide whether to overtake or not based on its surroundings. These results indicate that sparse dependency greatly models the situation that the noncontrollable dynamics are affected by the controllable dynamics, which is conducive to the downstream decision-making task by accurately predicting the noncontrollable dynamics at future moments.

Qualitative results. We show the video prediction results of Iso-Dream++ in the CARLA environment in Fig. 10. Because of the first-person view in this environment, the agent actions potentially affect all pixel values in the observation, as the camera on the main car (*i.e.*, the agent) moves. Therefore, we can view the dynamics of other vehicles as a combination of controllable and non-controllable states. Accordingly, our model determines which component is dominant by learning attention mask values between 0 and 1 across

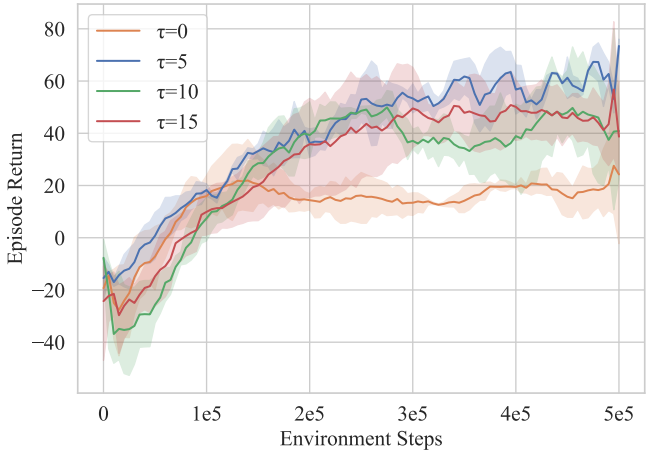


Fig. 11: The hyperparameter analysis of τ in CARLA.

the action-conditioned and action-free branches. The "action-free masks" present hot spots around other vehicles, while the attention values in corresponding areas on the "action-cond masks" are

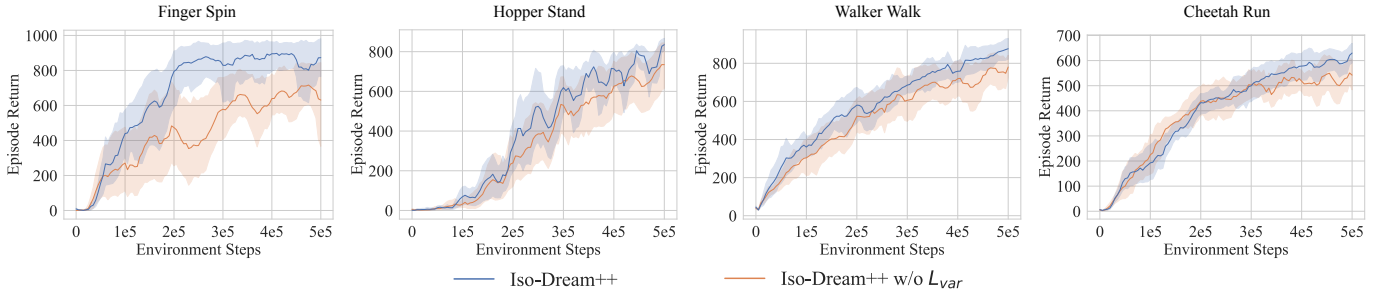


Fig. 12: The ablation study of the proposed variance constraints in DMC. We report the results averaged over 10 seeds.

TABLE 1: Qualitative results in DMC. The agents are trained and evaluated in environments with `video_easy` background. * indicates a different setup from that of DBC. Iso-Dream (*conf.*) is the model from our conference paper, which only uses the reconstruction loss (w/o KL divergence) in the action-free branch.

Method	Finger Spin	Hopper Stand	Walker Walk	Cheetah Run
SVEA	562 ± 22	6 ± 8	826 ± 65	178 ± 64
CURL	280 ± 50	451 ± 250	443 ± 206	269 ± 24
DBC*	1 ± 2	5 ± 9	32 ± 7	15 ± 5
DreamerV2	755 ± 92	260 ± 366	655 ± 47	475 ± 159
DreamerV3	124 ± 52	472 ± 328	701 ± 114	546 ± 117
DreamerPro	721 ± 147	295 ± 129	813 ± 88	297 ± 63
Denoised-MDP	635 ± 284	104 ± 117	214 ± 56	233 ± 119
Iso-Dream (*conf.*)	800 ± 59	746 ± 312	911 ± 50	$\mathbf{659 \pm 62}$
Iso-Dream	816 ± 16	769 ± 173	852 ± 97	597 ± 156
Iso-Dream++	$\mathbf{938 \pm 51}$	$\mathbf{877 \pm 34}$	$\mathbf{932 \pm 37}$	$\mathbf{639 \pm 19}$

still greater than zero. As shown in the third and fifth lines, Iso-Dream++ mainly learns the dynamics of mountains and trees in the action-conditioned branch and the dynamics of other driving vehicles in the action-free branch, respectively, which helps the agent avoid collisions by rolling out noncontrollable components to preview possible future states of other vehicles.

Hyperparameter analyses of τ. We evaluate the effect of using different numbers of rollout steps of the noncontrollable states as the inputs of the action model. From the results in Fig. 11, we observe that our model achieves the best performance at $\tau = 5$. However, there are no remarkable differences among $\tau \in [5, 10, 15]$, as long-term predictions for noncontrollable states may increase model errors. Besides, we implement a model without rolling out noncontrollable states into the future, *i.e.*, $\tau = 0$. It performs significantly worse than other baselines with $\tau \in [5, 10, 15]$, which demonstrates the benefit of rolling out the disentangled action-free branch in policy optimization.

4.3 DeepMind Control Suite

Implementation. We evaluate our model on `video_easy` and `video_hard` benchmarks from the DMC Generalization Benchmark [20], where the background is continually changing throughout an episode. All experiments use visual observations only, of shape $64 \times 64 \times 3$. The episodes last for 1,000 steps and have randomized initial states. We apply a fixed action repeat of $R = 2$ across tasks. In this environment, since the background is randomly replaced by a real-world video, the noncontrollable motion of the background will affect the procedure of dynamics learning and behavior learning. Therefore, to obtain a better decision policy and avoid the disruption from noisy backgrounds, the agent may decouple noncontrollable representation (*i.e.*, dynamic

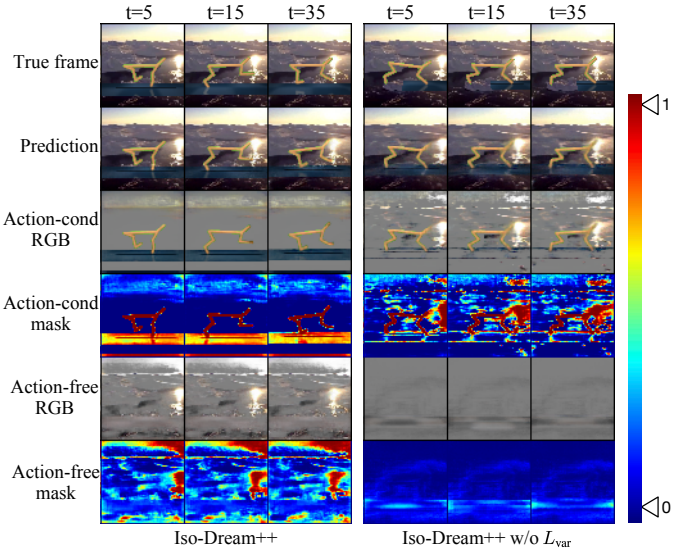


Fig. 13: Video prediction results from our approaches w/ and w/o the proposed variance constraints.

background) and controllable representation in spacetime, and only use controllable representation for control, thereby removing the modeling sparse dependency. Instead of training the action-free branch with only reconstruction loss in our preliminary work [1], we follow the structure described in Section 3.1 since the noncontrollable dynamics in some video backgrounds are complicated for learning, particularly `video_hard` benchmark. We evaluate the models using 4 tasks, *i.e.*, Finger Spin, Cheetah Run, Walker Walk, and Hopper Stand. The maximum number of environmental steps is $500k$. We use $\beta_1 = \beta_2 = 1$ and $\alpha = 1$ in Eq. (6) and $\lambda_1 = \lambda_2 = 1$ in Eq. (10).

Quantitative comparisons. We present the quantitative results of Iso-Dream++ for `video_easy` benchmark in Table 1. Our final model outperforms DreamerV2 and other baselines significantly in all tasks. Compared with DBC and Denoised-MDP, which both aim to extract task-relevant representation from complex visual distractions, our method is more powerful with large performance gains on all four tasks, indicating that disentangling different dynamics by modular structure and variance constraints provides more cleaner and useful information for the downstream task. Moreover, we have a better performance than DreamerPro, which is also based on Dreamer but learns the world model without reconstructing the observations. This demonstrates that our model effectively helps the agent to learn controllable visual representations and alleviate complex background interference.

Analyses of the min-max variance constraints. We investigate the effectiveness of the proposed variance constraints described

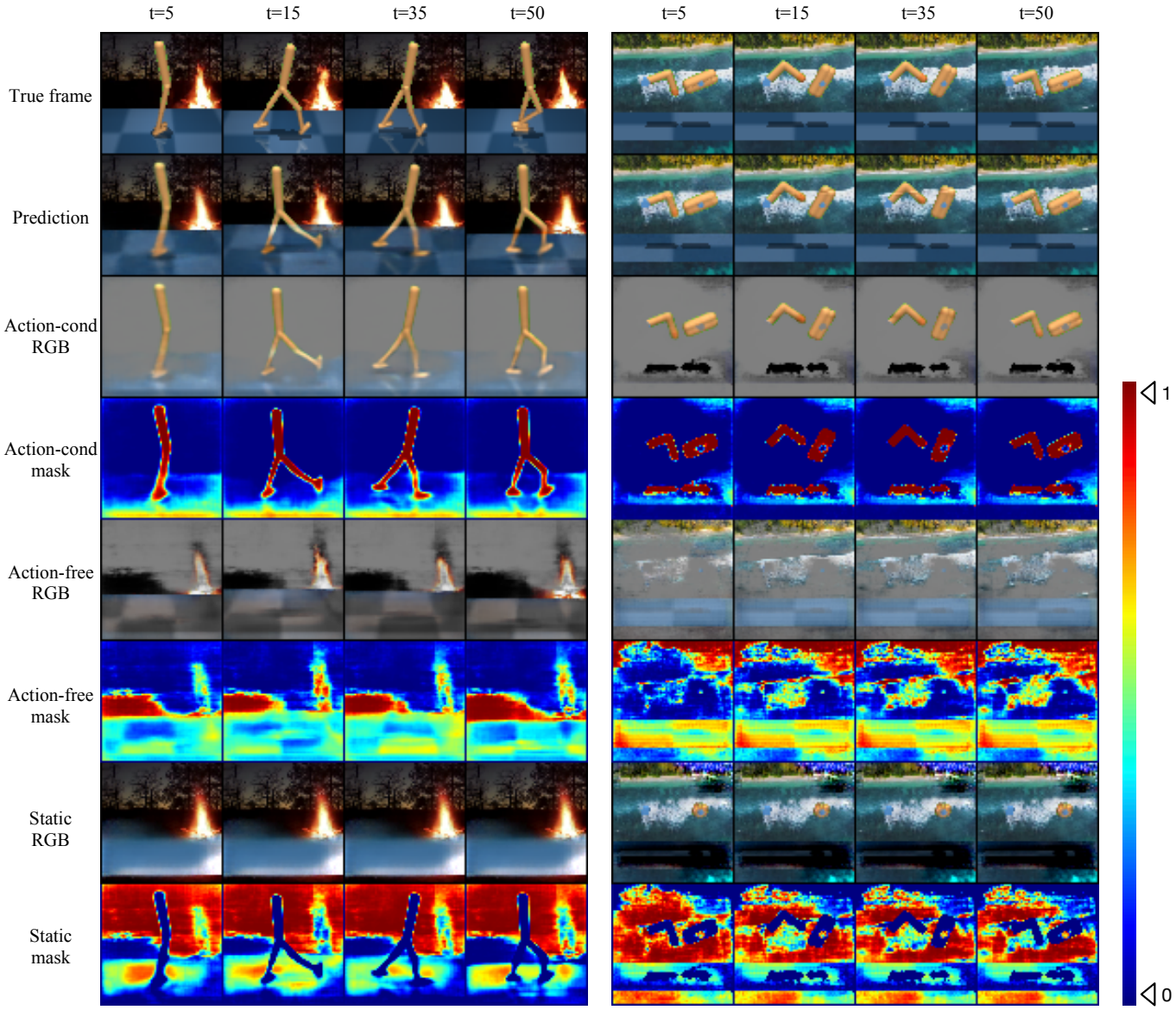


Fig. 14: Video prediction results with noisy backgrounds in DMC. For each sequence, we use the first 5 images as context frames. Iso-Dream++ successfully disentangles controllable and noncontrollable components.

in Section 3.1.2 by removing it from the training process of Iso-Dream++. As shown in Fig. 12, the compared models are trained for 10 seeds, and the proposed method improves the performance of our model in most tasks, especially in *finger spin*, where we have witnessed significant training collapse (see Fig. 4). In Fig. 13, we provide a qualitative comparison of the disentanglement results between models trained with and without variance constraints. Comparing the fifth and sixth row of action-free branch outputs, we observe that the action-free dynamics (such as the light over the lake) are correctly assigned to the action-free branch by variance constraints, preventing the action-conditioned branch from capturing all dynamic information, *i.e.*, training collapse. Because of the pure dynamics captured in the action-conditioned branch, our model with variance constraints gains definite improvements.

Qualitative results. We use Iso-Dream++ to perform video prediction in the DMC environment with `video_easy` backgrounds. The frame sequence and actions are randomly collected from test episodes. The first 5 frames are given to the model and the next 45 frames are predicted only based on action inputs. In this environment, the video background can be viewed as a combination of noncontrollable dynamics and static representations. Fig. 14 visualizes the entire generated RGB images, the decoupled RGB components, and the corresponding masks of the three network branches. From these results, we observe that our approach has the ability to predict long-term sequence and disentangle controllable

TABLE 2: The study of the generalization ability and robustness of the compared models to immediate visual distractions in DMC.

Method	Finger Spin	Hopper Stand	Walker Walk	Cheetah Run
Train: *video_easy*; Test: *video_hard*				
DreamerPro	628 ± 151	180 ± 96	533 ± 212	244 ± 27
Denoised-MDP	27 ± 21	44 ± 25	169 ± 61	103 ± 46
Iso-Dream++	**692** ± **185**	**643** ± **155**	**642** ± **129**	**441** ± **183**
Train: *video_easy*; Test: *video_easy with Gaussian noises*				
DreamerPro	663 ± 129	223 ± 76	824 ± 72	263 ± 55
Denoised-MDP	652 ± 306	103 ± 110	180 ± 79	195 ± 131
Iso-Dream++	**851** ± **109**	**806** ± **74**	**906** ± **31**	**582** ± **69**

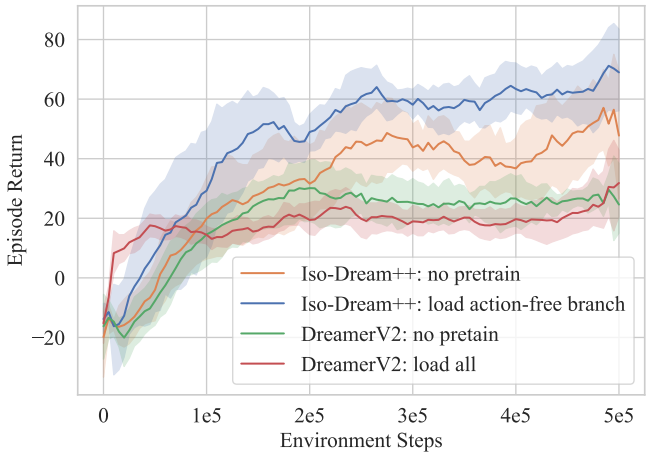


Fig. 15: Transfer learning results across `Day` and `Night` modes in DMC. Leveraging a pretrained *Day-mode* action-free branch can greatly benefit the finetuning results of Iso-Dream++ in the *Night mode*. Results are averaged over 10 seeds.

(agent) and noncontrollable dynamics (background motion) from complex visual images. As shown in the third and fourth rows in Fig. 14, the controllable representation has been successfully isolated and matches its mask. As shown in the fifth and sixth rows, the motion of fires and sea waves are captured as noncontrollable dynamics by the action-free branch.

Robustness to immediate distractions. To assess the ability of Iso-Dream++ to resist immediate visual distractions, we train the RL models on the *video_easy* benchmark and evaluate them on (i) *video_hard*; (ii) *video_easy with Gaussian noises*. In Table 2, we compare the results from Iso-Dream++ with those from DreamerPro and Denoised-MDP, which both focus on learning robust representations against visual noises. We observe a remarkable advantage of Iso-Dream++ against unexpected distractions, which consistently outperforms the DreamerPro and Denoised-MDP across all tasks.

4.4 Transfer Learning Analyses

Transfer of noncontrollable dynamics in CARLA. Our model learns different dynamics in different branches, which makes it naturally suitable for transfer learning. Unlike common methods that transfer all knowledge from a pretrained source task, we can selectively transfer specific knowledge for a target task. Specifically, we only transfer relevant knowledge, such as shared dynamics between source and target tasks, to achieve precise disentanglement and robust decision-making on the target task. In Fig. 7, We can see that the noncontrollable dynamics are similar between day and night modes, *i.e.*, the movement of other driving vehicles. We keep the action-free branch pretrained on the day mode of the CARLA environment and then train it on the night mode. The results are shown in Fig. 15. Comparing the orange curve and the blue curve, our model that transfers noncontrollable dynamics in the action-free branch has a significant improvement. However, the performance gain of DreamerV2 is small. Therefore, due to the modular structure in our Iso-Dream++, when there are two environments with similar dynamics, we can train on the easy environment first, and then load the specific pretrained branch to help the model learn on difficult tasks. Therefore, the modular structure of our Iso-Dream++ allows us to selectively transfer controllable or noncontrollable parts to novel domains based on our prior knowledge of the domain gap.

Transfer of controllable dynamics in DMC. For DMC, we use the `video_easy` (source) and `video_hard` (target) benchmarks to evaluate the transfer ability of our model. We transfer the controllable information in the action-conditioned branch because the controllable dynamics are the same in both environments, *i.e.*, the motion of the agent. From Fig. 16, we have two key observations. First, upon loading the pretrained action-conditioned branch, Iso-Dream++ exhibits a significant advantage over the non-pretrained counterpart. Second, it is noteworthy that the performance improvement achieved by our model through pretraining surpasses that of DreamerV2 by a considerable margin.

5 Related Work

Visual RL. In visual control tasks, RL agents learn the action policy directly from high-dimensional observations. They can be roughly grouped into two categories, that is, model-free methods [25, 23, 28, 29, 30, 24] and model-based methods [10, 5, 3, 11, 6, 9, 7, 26, 31, 27, 22, 32, 33]. Among them, the MBRL approaches explicitly model the state transitions and

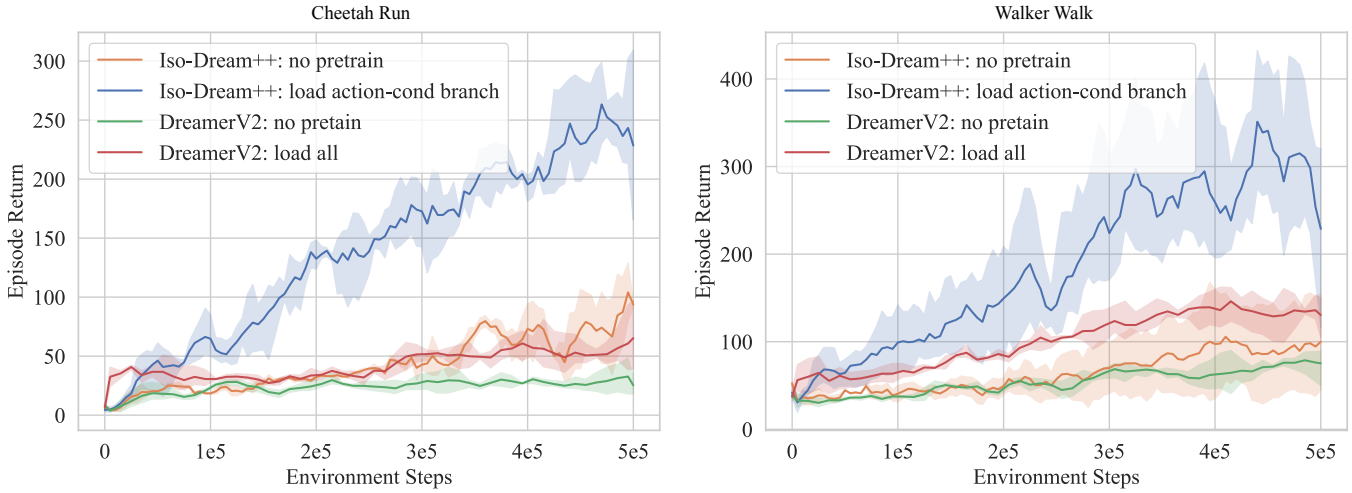


Fig. 16: Transfer learning results in DMC across environments with `video_easy` and `video_hard` backgrounds. Unlike DreamerV2, which can only transfer the entire world model (red curve) to the target domain, Iso-Dream++ enables us to separately transfer the pretrained action-conditioned branch and obtain significantly better finetuning results (blue curve).

generally yield higher sample efficiency than the model-free methods. A notable branch of work is the MuZero models, such as Stochastic MuZero [34]. These models simulate and explore possible future action trajectories through Monte Carlo tree search (MCTS), which can effectively improve long-term decision-making but introduce a vast computational cost. Notably, our model is different from Stochastic MuZero in two ways. First, we improve dynamics learning by encouraging representation decoupling, which we assume can enable the model to better understand the controllable and noncontrollable parts of the environment and greatly benefit the learned policy. Second, unlike in Stochastic MuZero, our model performs an actor-critic algorithm without MCTS, which is practical in short-term control tasks such as autonomous driving and ensures higher efficiency for both policy optimization and deployment. Another line of work is the so-called *World Models*. Ha and Schmidhuber [3] proposed to learn compressed latent states of the environment in a self-supervised manner and optimize potential behaviors based on the latent states generated by the world model. Similarly, PlaNet [11] introduces the *recurrent state-space model* (RSSM) as the world model and performs the cross-entropy method over the imagined recurrent states. DreamerV1-V3 [6, 8, 21] employ actor-critic methods to optimize the expected values and agent's behaviors over the predicted latent states in RSSM. Specifically, our model based on DreamerV2 outperforms DreamerV2-V3 remarkably in CARLA and DMC. We also note that the state decoupling and future-conditioned behavior learning techniques proposed in Iso-Dream++ can be seamlessly integrated with DreamerV2-V3, consistently enhancing their overall performance and convergence rate.

Visual RL with visual distractions. However, for complex visual environments with background or even dynamic distractions, it is still challenging to learn effective behavior policies. To tackle this problem, some approaches [26, 31, 27, 22] learn a more robust representation by discarding pixel-reconstruction to avoid struggling with the presence of visual noises. DreamerPro [22] uses online clustering to learn prototypes from the recurrent states of the world model, eliminating the need for reconstruction. Denoised-MDP [27] categorizes system dynamics into four types based on their controllability and relation to rewards, and optimizes the policy model only with information that is both controllable and relevant to rewards. It is worth noting that Iso-Dream++ differs significantly from the aforementioned methods in two key ways. First, we explicitly model the state transitions of controllable and noncontrollable dynamics in two distinct branches. This modular structure empirically facilitates transfer learning between related but distinct domains. Second, the decoupled world model offers a more versatile method of learning behavior. By previewing possible future states of noncontrollable patterns, we can make informed decisions at present. This also allows us to choose whether or not to incorporate noncontrollable states into our decision-making process, based on our prior knowledge of the specific domain.

Action-conditioned video prediction. Another branch of deep learning solutions to visual control problems is to learn action-conditioned video prediction models [2, 35, 36, 37, 38] and then perform Monte-Carlo importance sampling and optimization algorithms, such as the *cross-entropy methods*, over available behaviors [10, 4, 39]. Hot topics in video prediction mainly include long-term and high-fidelity future frames generation [40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52], dynamics uncertainty modeling [53, 54, 55, 56, 57, 58, 59], object-centric scene decomposition [60, 61, 62, 63, 64, 65, 66], and space-time disentanglement [67, 61, 68, 69]. The corresponding technical improvements mainly involve the use of more effective neural architectures, novel probabilistic modeling methods, and specific forms of video representations. The disentanglement methods are closely related to the world model in Iso-Dream++. They commonly separate visual dynamics into content and motion vectors, or long-term and short-term states. In contrast, Iso-Dream++ is designed to learn a decoupled world model based on controllability, which contributes more to the downstream behavior learning process.

6 Conclusion

In this paper, we proposed an MBRL framework named Iso-Dream++, which mainly tackles the difficulty of vision-based prediction and control in the presence of complex visual dynamics. Our approach has four novel contributions to world model representation learning and corresponding MBRL algorithms. First, it learns to decouple controllable and noncontrollable latent state transitions via modular network structures and inverse dynamics. Second, it introduces the min-max variance constraints to prevent "training collapse", where a single state transition branch captures all information. Third, it makes long-horizon decisions by rolling out the noncontrollable dynamics into the future and learning their influences on current behavior. Fourth, it models the sparse dependency of future noncontrollable dynamics on current controllable dynamics to deal with some practical dynamic environments. Iso-Dream++ achieves competitive results on the CARLA autonomous driving task, where other vehicles can be naturally viewed as noncontrollable components, indicating that with the help of decoupled latent states, the agent can make more forward-looking decisions by previewing possible future states in the action-free network branch. Besides, Our approach was shown to effectively improve the visual control task in a modified DeepMind Control Suite, achieving significant advantages over existing methods in standard, noisy, and transfer learning setups.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62250062, 62106144, U19B2035), the Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0102), the Fundamental Research Funds for the Central Universities, and the Shanghai Sailing Program (Grant No. 21Z510202133).

References

[1] M. Pan, X. Zhu, Y. Wang, and X. Yang, "Iso-dream: Isolating and leveraging noncontrollable visual dynamics in world models," in *NeurIPS*, 2022, pp. 23 178–23 191.

[2] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," in *NeurIPS*, 2015, pp. 2863–2871.

[3] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *NeurIPS*, 2018, pp. 2455–2467.

[4] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

[5] J. Oh, S. Singh, and H. Lee, "Value prediction network," in *NeurIPS*, 2017, pp. 6118–6128.

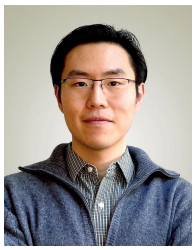
[6] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," in *ICLR*, 2020.
[7] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak, "Planning to explore via self-supervised world models," in *ICML*, 2020, pp. 8583–8592.
[8] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," in *ICLR*, 2020.
[9] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine *et al.*, "Model-based reinforcement learning for Atari," in *ICLR*, 2020.
[10] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *ICRA*, 2017, pp. 2786–2793.
[11] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *ICML*, 2019, pp. 2555–2565.
[12] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *ICML*, 2019, pp. 4114–4124.
[13] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020, pp. 9729–9738.
[14] Q. Qian, Y. Xu, J. Hu, H. Li, and R. Jin, "Unsupervised visual representation learning by online constrained k-means," in *CVPR*, 2022, pp. 16 640–16 649.
[15] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, "Object-centric learning with slot attention," in *NeurIPS*, 2020, pp. 11 525–11 538.
[16] A. Goyal, A. Lamb, J. Hoffmann, S. Sodhani, S. Levine, Y. Bengio, and B. Schölkopf, "Recurrent independent mechanisms," in *ICLR*, 2021.
[17] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *TNN*, pp. 1054–1054, 1998.
[18] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "CARLA: an open urban driving simulator," in *CoRL*, 2017, pp. 1–16.
[19] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq *et al.*, "Deepmind control suite," *arXiv preprint arXiv:1801.00690*, 2018.
[20] N. Hansen and X. Wang, "Generalization in reinforcement learning by soft data augmentation," in *ICRA*, 2021, pp. 13 611–13 617.
[21] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.
[22] F. Deng, I. Jang, and S. Ahn, "DreamerPro: Reconstruction-free model-based reinforcement learning with prototypical representations," in *ICML*, 2022, pp. 4956–4975.
[23] M. Laskin, A. Srinivas, and P. Abbeel, "CURL: contrastive unsupervised representations for reinforcement learning," in *ICML*, 2020, pp. 5639–5650.
[24] N. Hansen, H. Su, and X. Wang, "Stabilizing deep q-learning with convnets and vision transformers under data augmentation," in *NeurIPS*, 2021, pp. 3680–3693.
[25] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.
[26] A. Zhang, R. McAllister, R. Calandra, Y. Gal, and S. Levine, "Learning invariant representations for reinforcement learning without reconstruction," in *ICLR*, 2021.
[27] T. Wang, S. S. Du, A. Torralba, P. Isola, A. Zhang, and Y. Tian, "Denoised mdps: Learning world models better than the world itself," in *ICML*, 2022, pp. 22 591–22 612.
[28] D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus, "Improving sample efficiency in model-free reinforcement learning from images," in *AAAI*, 2021, pp. 10 674–10 681.
[29] D. Yarats, I. Kostrikov, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," in *ICLR*, 2021.
[30] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, "Reinforcement learning with augmented data," in *NeurIPS*, 2020, pp. 19 884–19 895.
[31] H. Bharadhwaj, M. Babaeizadeh, D. Erhan, and S. Levine, "Information prioritization through empowerment in visual model-based RL," in *ICLR*, 2022.
[32] Y. Xu, J. Parker-Holder, A. Pacchiano, P. J. Ball, O. Rybkin, S. J. Roberts, T. Rocktäschel, and E. Grefenstette, "Learning general world models in a handful of reward-free deployments," in *NeurIPS*, 2022, pp. 26 820–26 838.
[33] T. Ji, Y. Luo, F. Sun, M. Jing, F. He, and W. Huang, "When to update your model: Constrained model-based reinforcement learning," in *NeurIPS*, 2022, pp. 23 150–23 163.
[34] I. Antonoglou, J. Schrittwieser, S. Ozair, T. K. Hubert, and D. Silver, "Planning in stochastic environments with a learned model," in *ICLR*, 2022.
[35] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *NeurIPS*, 2016, pp. 64–72.
[36] S. Chiappa, S. Racaniere, D. Wierstra, and S. Mohamed, "Recurrent environment simulators," in *ICLR*, 2017.
[37] Y. Wang, H. Wu, J. Zhang, Z. Gao, J. Wang, S. Y. Philip, and M. Long, "Predrnn: A recurrent neural network for spatiotemporal predictive learning," *TPAMI*, pp. 2208–2225, 2022.
[38] M. Babaeizadeh, M. T. Saffar, S. Nair, S. Levine, C. Finn, and D. Erhan, "Fitvid: Overfitting in pixel-level video prediction," *arXiv preprint arXiv:2106.13195*, 2021.
[39] M. Jung, T. Matsumoto, and J. Tani, "Goal-directed behavior under variational predictive coding: Dynamic organization of visual attention and working memory," in *IROS*, 2019, pp. 1040–1047.
[40] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *ICML*, 2015, pp. 843–852.
[41] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *NeurIPS*, 2015, pp. 802–810.
[42] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *NeurIPS*, 2016, pp. 613–621.
[43] P. Bhattacharjee and S. Das, "Temporal coherency based criteria for predicting video frames using deep multi-stage generative adversarial networks," in *NeurIPS*, 2017, pp. 4271–4280.
[44] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "Predrnn: Recurrent neural networks for predictive learning using

spatiotemporal lstms," in *NeurIPS*, 2017, pp. 879–888.
[45] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *ICML*, 2017, pp. 3560–3569.
[46] N. Wichers, R. Villegas, D. Erhan, and H. Lee, "Hierarchical long-term video prediction without supervision," in *ICML*, 2018, pp. 6038–6046.
[47] F. A. Reda, G. Liu, K. J. Shih, R. Kirby, J. Barker, D. Tarjan, A. Tao, and B. Catanzaro, "Sdc-net: Video prediction using spatially-displaced convolution," in *ECCV*, 2018, pp. 718–733.
[48] M. Oliu, J. Selva, and S. Escalera, "Folded recurrent neural networks for future video prediction," in *ECCV*, 2018, pp. 716–731.
[49] W. Liu, A. Sharma, O. Camps, and M. Sznaier, "Dyan: A dynamical atoms-based network for video prediction," in *ECCV*, 2018, pp. 170–185.
[50] J. Xu, B. Ni, Z. Li, S. Cheng, and X. Yang, "Structure preserving video prediction," in *CVPR*, 2018, pp. 1460–1469.
[51] B. Jin, Y. Hu, Q. Tang, J. Niu, Z. Shi, Y. Han, and X. Li, "Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction," in *CVPR*, 2020, pp. 4554–4563.
[52] N. Behrmann, J. Gall, and M. Noroozi, "Unsupervised video representation learning by bidirectional feature prediction," in *WACV*, 2021, pp. 1670–1679.
[53] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," in *ICLR*, 2018.
[54] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," in *ICML*, 2018, pp. 1174–1183.
[55] R. Villegas, A. Pathak, H. Kannan, D. Erhan, Q. V. Le, and H. Lee, "High fidelity video prediction with large stochastic recurrent neural networks," in *NeurIPS*, 2019, pp. 81–91.
[56] T. Kim, S. Ahn, and Y. Bengio, "Variational temporal abstraction," in *NeurIPS*, 2019, pp. 11 570–11 579.
[57] L. Castrejon, N. Ballas, and A. Courville, "Improved conditional VRNNs for video prediction," in *ICCV*, 2019, pp. 7608–7617.
[58] J.-Y. Franceschi, E. Delasalles, M. Chen, S. Lamprier, and P. Gallinari, "Stochastic latent residual video prediction," in *ICML*, 2020, pp. 3233–3246.
[59] B. Wu, S. Nair, R. Martin-Martin, L. Fei-Fei, and C. Finn, "Greedy hierarchical variational autoencoders for large-scale video prediction," in *CVPR*, 2021, pp. 2318–2328.
[60] S. van Steenkiste, M. Chang, K. Greff, and J. Schmidhuber, "Relational neural expectation maximization: Unsupervised discovery of objects and their interactions," in *ICLR*, 2018.
[61] J.-T. Hsieh, B. Liu, D.-A. Huang, L. F. Fei-Fei, and J. C. Niebles, "Learning to decompose and disentangle representations for video prediction," in *NeurIPS*, 2018, pp. 517–526.
[62] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner, "Multi-object representation learning with iterative variational inference," in *ICML*, 2019, pp. 2424–2433.
[63] P. Zablotskaia, E. A. Dominici, L. Sigal, and A. M. Lehrmann, "Unsupervised video decomposition using spatio-temporal iterative inference," *arXiv preprint arXiv:2006.14727*, 2020.
[64] X. Bei, Y. Yang, and S. Soatto, "Learning semantic-aware dynamics for video prediction," in *CVPR*, 2021, pp. 902–912.
[65] K. Greff, S. van Steenkiste, and J. Schmidhuber, "Neural expectation maximization," in *NeurIPS*, 2017, pp. 6694–6704.
[66] A. R. Kosiorek, H. Kim, I. Posner, and Y. W. Teh, "Sequential attend, infer, repeat: generative modelling of moving objects," in *NeurIPS*, 2018, pp. 8615–8625.
[67] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," in *ICLR*, 2017.
[68] V. L. Guen and N. Thome, "Disentangling physical dynamics from unknown factors for unsupervised video prediction," in *CVPR*, 2020, pp. 11 474–11 484.
[69] N. Bodla, G. Shrivastava, R. Chellappa, and A. Shrivastava, "Hierarchical video prediction using relational layouts for human-object interactions," in *CVPR*, 2021, pp. 12 146–12 155.

Minting Pan received the B.E. degree from Hunan University in 2018. She is currently pursuing her PhD degree in Shanghai Jiao Tong University. Her research interests lie on the model-based decision reinforcement learning, especially visual control tasks.

Xiangming Zhu received the B.E. degree from Shanghai Jiao Tong University in 2022. He is currently pursuing his Master degree in Shanghai Jiao Tong University. His research interests lie on the intersection of machine learning and computer vision, especially vision-based intuitive physics and reinforcement learning.

Yitao Zheng received the B.E. degree from Xidian University in 2023. He is currently purchasing his Ph.D degree in Shanghai Jiao Tong University. His research interest lies in the intersection of computer vision and deep learning, especially model-based visual reinforcement learning.



Yunbo Wang received the B.E. degree from Xi'an Jiaotong University in 2012, and the M.E. and Ph.D. degrees from Tsinghua University in 2015 and 2020. He received the CCF Outstanding Doctoral Dissertation Award in 2020, advised by Philip S. Yu and Mingsheng Long. He is now an assistant professor at the AI Institute and the Department of Computer Science at Shanghai Jiao Tong University. He does research in deep learning, especially predictive learning, spatiotemporal modeling, and model-based decision making.

Xiaokang Yang received the B.S. degree from Xiamen University in 1994, the M.S. degree from the Chinese Academy of Sciences in 1997, and the Ph.D. degree from Shanghai Jiao Tong University in 2000. He is currently a Distinguished Professor, Shanghai Jiao Tong University, Shanghai, China. His current research interests include visual signal processing and communication, media analysis and retrieval, and pattern recognition. He serves as an Associate Editor of IEEE Transactions on Multimedia.